HW 1

Hey Igor! I've been playing with Asciidoctor lately. GitHub supports it (but not fully). I'm not sure any of the features are useful for this task, but it's been awesome to use to create pdfs. I've even been using it for really simple slide shows. All through emacs! No more dragging, aligning, changing font sizes. The beauty is that you can type all your documents in markup, and can output to PDF or HTML with really nice defaults (linked to output). The Quick Reference shows the syntax. I've really been geeking out over it lately! Back to the HW!

Problem 1

Source (notice how this title renders in PDF and HTML)

```
from math import hypot

def pythagoreanTheorem(length_a, length_b):
    return hypot(length_a, length_b)


print(pythagoreanTheorem(2, 2))
print(pythagoreanTheorem(3, 2))
print(pythagoreanTheorem(2, 4))
```

Output (it also supports some markdown syntax (```))

```
2.8284271247461903
3.6055512754639896
4.47213595499958
```

Problem 2

Source

```
def list_mangler(list_in):
    list_out = []

    for item in list_in:
        # even
        if (item % 2 == 0):
            list_out.append(item * 2)
        # odd
        else:
            list_out.append(item * 3)

    return list_out
```

Output

```
print(list_mangler([1, 2, 3, 4]))
print(list_mangler([5, 6, 7, 8]))
print(list_mangler([9, 10, 11, 110, 111]))
```

```
[3, 4, 9, 8]
[15, 12, 21, 16]
[27, 20, 33, 220, 333]
```

Explanation

A number is even if it's divisible by 2, odd otherwise.

Alternative way is interesting but hurts my head

```
squares = [x**2 for x in range(10)]
```

Problem 3

Source

```
from statistics import mean

def grade_calc(grades_in, to_drop):
    # leave original list unmodified
    grades_out = grades_in.copy()

    # sort and drop lowest `to_drop`
    grades_out.sort()

    for iter in range(to_drop):
        grades_out.pop(0)

    # calc avg of modified list
    avg = mean(grades_out)

    # give letter grade
    if (100 >= avg >= 90):
        return 'A'

    elif (89 >= avg >= 80):
        return 'B'

    elif (79 >= avg >= 70):
        return 'C'

    elif (69 >= avg >= 60):
        return 'D'

    else:
        return 'F'
```

Output

```
print(grade_calc([100, 90, 80, 95], 2))
print(grade_calc([0, 90, 80, 95], 2))
print(grade_calc([70, 100, 60, 50], 2))
```

```
A
A
B
```

Explanation

I broke the problem up by commenting out what it needs to do. Then I filled in the code. The comments remain. I used the `sort()` function and the `pop()` function to remove the lowest. I made a copy of the list before doing so.

Problem 4

Source

```
def odd_even_filter(numbers):
    # separate odd and even
    even = []
    odd = []

    for num in numbers:
        if (num % 2 == 0):
            even.append(num)

        else:
            odd.append(num)

    return [even, odd]
```

Output

```
print(odd_even_filter([1, 2, 3, 4, 5, 6, 7, 8, 9]))
print(odd_even_filter([3, 9, 43, 7]))
print(odd_even_filter([71, 39, 98, 79, 5, 89, 50, 90, 2, 56]))
```

```
[[2, 4, 6, 8], [1, 3, 5, 7, 9]]
[[], [3, 9, 43, 7]]
[[98, 50, 90, 2, 56], [71, 39, 79, 5, 89]]
```

Explanation

At first I had two loops

```
# make list of all even
even = []

for num in numbers:
    if (num % 2 == 0):
        even.append(num)

# make list of all odd
odd = []

for num in numbers:
    if (num % 2 == 1):
        odd.append(num)
```

but then I realized I could use the same loop with an else.

I also realized I could simplify the return statement to one line

```
output = []
output.append(even)
output.append(odd)

return output
```